

Bericht zum Modul Praxis II
Praxisphase 3: 04.01. - 29.03.2010
Praxisphase 4: XX.XX. - XX.XX.2010

**Anbindung
eines Temperatur- und Luftdrucksensors
an einen „System on Chip-Webserver“
über SPI-Bus als Fortführung der Arbeit
aus den ersten beiden Praxisphasen**

von

Christian König

Matrikelnr.: 207313

Kurs: TIT08AGR



Deutsches Zentrum
für Luft- und Raumfahrt e.V.
in der Helmholtz-Gemeinschaft

Standort Braunschweig

Institut für Flugsystemtechnik
Abteilung Hubschrauber

organisatorisch: Martin Gestwa
fachlich: Dipl.-Ing. Michael Przybilla

Ehrenwörtliche Erklärung

Hiermit versichere ich, dass ich
die vorliegende Arbeit selbstständig
und nur unter Verwendung der als Quellen
angewiesenen Hilfsmittel angefertigt habe.

Braunschweig, den XX.XX.XXXX _____
Christian König

Inhaltsverzeichnis

Ehrenwörtliche Erklärung	I
Abkürzungsverzeichnis	IV
Abbildungsverzeichnis	V
Listings	V
Vorwort	VI
1 Einleitung	1
1.1 Abteilung Hubschrauber - Gruppe Rotorversuchstand	1
1.2 Aufgabenstellung	2
2 Grundlagen	4
2.1 Mikrocontroller	4
2.2 Programmiersprache C	4
2.3 Übertragungsprotokolle	5
2.3.1 Ethernet 10/100Base-T Halbduplex	5
2.3.2 Serial Peripheral Interface	6
3 Verwendete Hardware	7
3.1 Mikrocontroller <i>SC13</i> von Beck-IPC	7
3.2 Entwicklungsboard <i>DK50</i> von Beck-IPC	8
3.3 Drucksensor <i>SCP1000-D01</i> von VTI	9
3.4 Display <i>GU128x18F-K612A4</i> von Noritake Itron	10
4 Verwendete Software	12
4.1 Borland C++ und Borland C++ Compiler	12
4.2 CHIPtool	12
4.3 Mozilla Firefox	12
4.4 Institutseigene Software	13
4.4.1 Extserver	13
4.4.2 Quickview	13

5	Anbindung der Peripherie-Geräte	14
5.1	Ethernet	14
5.2	Stromversorgung	14
5.3	Luftfeuchtesensor	14
5.4	Druck- und Temperatursensor	16
5.4.1	Aufbau der Hardware	16
5.4.2	Entwicklung der Software	16
5.5	Display	16
5.5.1	Aufbau der Hardware	16
5.5.2	Entwicklung der Software	16
6	Tests	19
6.1	Funktionstest	19
6.2	Dauertest	19
7	Ergebnisse der Arbeit	20
8	Fazit	21
	Literaturverzeichnis	A
	Anhang	D
	Beschriftung des DK50 [3]	E
	Blockschaltbild des DK50 [3]	F
	Schaltplan des DK50 [3]	G
	Datenblatt des SC13 [1]	I

Abkürzungsverzeichnis

CS	C hip- S elect
CSMA	C arrier S ense M ultiple A ccess/ C ollision D etection
DHCP	D ynamic H ost C onfiguration P rotocol
DLR	D eutsches Zentrum für L uft- und R aumfahrt
DNS	D omain N ame S ystem
DNW	D eutsch- N iederländische W indkanäle
EIA	E lectronic I ndustries A lliance
FHS	F lying H elicopter S imulator
FTP	F ile T ransfer P rotocol
HTML	H ypertext M arkup L anguage
I ² C	I nter I ntegrated C ircuit
IEEE	I nstitute of E lectrical and E lectronics E ngineers
LLF	L arge L ow-Speed F acility
MAC	M edia A ccess C ontrol
MISO	M aster I n S lave O ut
MOSI	M aster O ut S lave I n
MUM-T	M anned and U n M anned T eaming
PCB	P rinted C ircuit B oard
PIO	P rogrammable I nput/ O utput
RS	R adio S ector
RTOS	R ea L - T ime O perating S ystem
SCK	S erial C lock
SMD	S urface- M ounted D evice
SPI	S erial P eripheral I nterface
SS	S lave- S elect
Telnet	T elecommunication N etwork
VFD	V acuum F luorescent D isplay

Abbildungsverzeichnis

1	DLR - Forschung für die Erde [15]	VI
2	Flying Helicopter Simulator FHS [14]	1
3	Steuerpult des Rotorversuchstandes (links) [13] und Rotor Test Stand . . .	2
4	Sternförmige SPI-Busstruktur mit drei Slaves [11]	6
5	Mikrocontroller SC13 [2]	7
6	Entwicklungsboard DK50 ohne Mikrocontroller [3]	8
7	Druck- und Temperatursensor SCP1000 von VTI in Detailansicht und auf dem Test PCB Carrier [29] [31]	9
8	Display GU128x18F-K612A4 [19]	11
9	Institutseigene Software Extserver	13
10	Teilschaltbild mit Mikrocontroller SC13 und Sensor SCP1000	17
11	Teilschaltbild mit Mikrocontroller SC13 und Display	17
12	Schaltbild mit Mikrocontroller SC13, Display und Sensor SCP1000	18

Tabellenverzeichnis

1	Messmodi des Sensors SCP1000 [30]	10
2	molare Wärmekapazität bei konstantem Volumen	15
3	molare Wärmekapazität	15
4	Zusammensetzung trockener Luft	15

Listings

Vorwort

Das Deutsche Zentrum für Luft- und Raumfahrt ist das Forschungszentrum der Bundesrepublik Deutschland für die Bereiche Luftfahrt, Raumfahrt, Verkehr und Energie. Es fungiert als Raumfahrtagentur der Bundesregierung und kooperiert in diesem Bereich, genau wie in der Forschung, national und international. So betreibt das DLR beispielsweise das Kontrollzentrum des Columbus-Labors der ISS. Die Zielsetzung der Forschung ist in der Internetpräsenz des DLR [15] gut zusammengefasst:

„Die Mission des DLR umfasst (...)

- die Erforschung von Erde und Sonnensystem,
- die Forschung für den Erhalt der Umwelt,
- die Entwicklung umweltverträglicher Technologien zur Steigerung der Mobilität sowie für Kommunikation und Sicherheit.“

Das DLR betreibt in Deutschland 29 Institute bzw. Test- und Betriebseinrichtungen, die sich auf 13 Standorte verteilen. International werden Büros in Brüssel, Paris und Washington D.C. unterhalten. Insgesamt beschäftigt das DLR etwa 6200 Mitarbeiter.

Die vorliegende Arbeit wurde im Institut Flugsystemtechnik in der Abteilung Hubschrauber erstellt. Die Abteilung befasst sich mit Rotorsystem- und Flugsystemtechnik, mit dem Ziel, das Fliegen sicherer, wirtschaftlicher und umweltfreundlicher zu machen.



Abbildung 1: DLR - Forschung für die Erde [15]

1 Einleitung

1.1 Abteilung Hubschrauber - Gruppe Rotorversuchstand

Die Abteilung Hubschrauber des Instituts für Flugsystemtechnik befasst sich mit der Rotorsystem- und Flugsystemtechnik an Drehflüglern im Allgemeinen und Hubschraubern im Speziellen.

Hierbei ist der Fokus wie folgt gewählt:

„Zielrichtung der Aktivitäten ist die Unterstützung der Entwicklung neuer Technologien und deren Bewertung im Hinblick auf die Verbesserung der Flugeigenschaften, der Sicherheit, der Wirtschaftlichkeit, der Umweltverträglichkeit und der Missionseffektivität für die Fluggeräte.“ [12]

Um diese Ziele zu erreichen werden Verfahren eingesetzt wie Systemanalyse, Modellierung von Gesamt- und Teilsystemen, Windkanalversuche und schließlich die Erprobung im Flug.

Aktuelle Projekte sind unter Anderem die Entwicklung von Unterstützungssystemen für den Hubschrauber Allwetterflug oder für den „easy handling“ Hubschrauber, bei dem der Pilot beispielsweise durch aktive Steuerelemente fühlbar über Systemzustände informiert und so über eine Annäherung an Systemgrenzen gewarnt wird. Dieses Verfahren wird im FHS getestet, dem in Europa einzigartigem Forschungs- und Versuchshubschrauber, mit dem die Erprobung neuer Techniken im realen Flug möglich ist. [12]



Abbildung 2: Flying Helicopter Simulator FHS [14]

Die Abteilung Hubschrauber ist auch in der Kooperation tätig, wie zum Beispiel mit der Abteilung *Unbemannte Luftfahrzeuge*. Ein aktuelles Projekt befasst sich mit dem gemeinsamen Einsatz bemannter und unbemannter Luftfahrzeuge, bei dem ein unbemannter Hubschrauber als Aufklärungsdrohne für einen bemannten Hubschrauber fungiert. Für

den gemeinsamen Einsatz beider Hubschrauber im selben Luftraum müssen prozedurale und technische Verfahren entwickelt werden, wie zum Beispiel die Mensch-Maschine-Schnittstellen oder der Automatisierungsgrad des unbemannten Hubschraubers.

Die Entwicklung neuer Rotorsysteme bzw. die Modifikation bestehender wird mit dem Ziel vorangetrieben, Vibrationen und Lärm zu verringern und gleichzeitig die Leistungsfähigkeit zu erhöhen.

Die in Berechnungsverfahren entworfenen Rotoren müssen mit Hubschraubermodellen im Windkanal überprüft werden. Im Rotorversuchstand im DLR Braunschweig werden isolierte Haupt- und Heckrotoren, aber auch Gesamtsysteme getestet, bevor diese im DNW-LLF, einem Niedergeschwindigkeitswindkanal, endgültig validiert werden. Getestet werden dynamisch skalierte Rotoren, die einen Durchmesser von bis zu 4.2 Metern aufweisen. Durch die Größe, die etwa 40% eines Originalrotors beträgt, lassen sich die Messergebnisse leicht auf die Originalgröße übertragen.

Im Rotorversuchstand stehen zwei Teststände zur Verfügung, das *Modulare Windkanal Modell* und der *Rotor Test Stand*. Beide Teststände verfügen über eine 6-Komponenten-Rotorwaage, mit denen alle am Rotorkopf auftretenden Kräfte bestimmt werden können. Insgesamt besteht das Datenerfassungssystem aus bis zu 350 Kanälen, die Informationen aus dem drehenden und dem festen System liefern. [13]

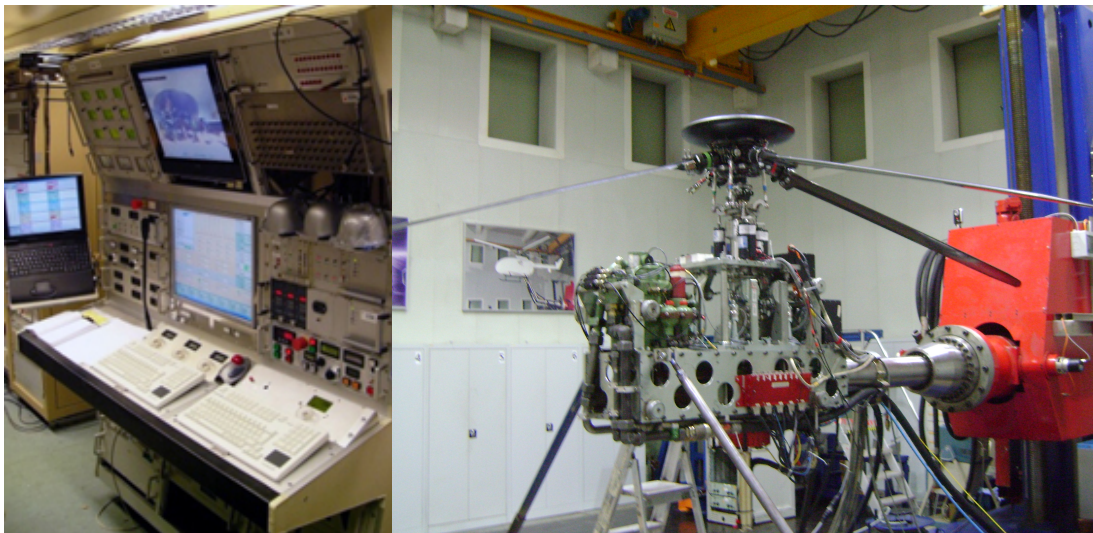


Abbildung 3: Steuerpult des Rotorversuchstandes (links) [13] und Rotor Test Stand

1.2 Aufgabenstellung

Wie bereits erwähnt, ist es im Entwicklungsprozess notwendig, die aerodynamischen Eigenschaften neuer Komponenten oder Gesamtsysteme im Windkanal zu testen, um die Berechnungen zu validieren. Die abschließenden Tests für Entwicklungen des DLR, die dem Bereich Hubschrauberrotoren zugehören, werden in dem deutsch-niederländischen Niedergeschwindigkeitswindkanal DNW-LLF in Marknesse in den Niederlanden durchgeführt.

Während der Tests werden die Umgebungsbedingungen im Windkanal gemessen und an einen Datenserver geschickt, um eine optimale Auswertung zu ermöglichen. Zu den Messungen gehören die Bestimmung des Drucks an drei verschiedenen Stellen, der Temperatur und der Luftfeuchte.

Bei diesen Versuchen müssen die wissenschaftlichen Mitarbeiter vor Ort sein und das Rotormodell, das einen Durchmesser von bis zu 4.2 m hat, muss unbeschadet von Braunschweig nach Marknesse und wieder zurück transportiert werden. Hieraus ergibt sich logischerweise ein hoher Zeit- und Kostenaufwand. Um diesen zu minimieren, werden die Windkanaltests in Braunschweig im Rotorversuchstand vorbereitet. Um die Tests ebenso auswerten zu können wie die Tests im DNW-LLF soll nun auch die Messung der Umgebungsbedingungen im Rotor Test Stand erfolgen und ebenfalls an einen Datenserver übertragen werden.

Aus den gemessenen Umgebungsbedingungen sollen weitere Daten berechnet, wie beispielsweise die Schallgeschwindigkeit oder die Luftdichte. Das System soll über eine Weboberfläche konfigurierbar sein, auf der zum Beispiel die aktuelle Einstellung der Rotorblätter eingetragen werden kann, sodass diese mit an den Datenserver übertragen wird. Ebenfalls soll es möglich sein, auf dieser Weboberfläche das Senden der Daten an den Server zu aktivieren bzw. zu deaktivieren.

Zusätzlich soll ein Display integriert werden, auf dem die aktuelle Temperatur und der aktuelle Luftdruck angezeigt werden sowie eine Information darüber, ob gerade Daten an den Server gesendet werden.

Die Implementierung der Weboberfläche sowie des Mechanismus zum Senden der Daten an den Server wurden bereits in der zweiten Praxisphase durchgeführt. Die Aufgabe für diese Praxisphase ist also, das Display und den Satz Sensoren anzubinden und die Daten der Sensoren auszuwerten. Weiterhin soll das Messsystem in ein Gehäuse integriert werden.

2 Grundlagen

2.1 Mikrocontroller

Unter dem Begriff Mikrocontroller versteht man ein eigenständiges Rechnersystem, dass vollständig bzw. nahezu vollständig in einem Halbleiterchip untergebracht ist. Im Gegensatz zu Mikroprozessoren sind so in diesem Chip auch Peripheriefunktionen untergebracht, wie zum Beispiel Analog-Digital-Wandler, Schnittstellenbausteine und Interruptcontroller. Schnittstellen können beispielsweise I²C, SPI, Ethernet oder serielle Schnittstellen sein.

In vielen Fällen sind neben den Peripheriefunktionen auch Arbeits- und Programmspeicher komplett oder teilweise in den Chip integriert. Als Speicher wird in der Regel Flash verwendet, da dieser ein wiederholtes Aufspielen von Programm-Code in der Entwicklungsphase ermöglicht.

Bei den Prozessoren ist vor allem ein niedriger Stromverbrauch wichtig. Von 4-Bit-Architekturen bis 32-Bit-Architekturen ist alles vertreten.

Die am häufigsten für Mikrocontroller verwendeten Programmiersprachen sind C und Assembler, aber auch andere Sprachen sind möglich. Welche Sprache verwendet werden sollte, hängt unter anderem von der Architektur des Mikrocontrollers ab. So wird bei 4-Bit- und 8-Bit-Architekturen hauptsächlich Assembler verwendet, wohingegen bei 16-Bit- und 32-Bit-Architekturen häufig Hochsprachen wie C eingesetzt werden.

Mikrocontroller sind in vielen elektrischen Geräten verbaut, die im Alltag genutzt werden. Beispielsweise in Waschmaschinen, Fernbedienungen oder im Auto werden Mikrocontroller eingesetzt. Im Auto werden zum Beispiel Airbag, Motor oder Klimaanlage auf diese Weise gesteuert. Auch der Chip auf einer EC-Karte ist ein Mikrocontroller. [26]

2.2 Programmiersprache C

Die Programmiersprache C wurde 1972 von Dennis Ritchie entwickelt und zählt zu den Hochsprachen. C ermöglicht es also, Programme in einer abstrakten Sprache zu verfassen. Da diese Sprache für Computer nicht lesbar ist, ist eine Übersetzung in Maschinensprache notwendig, bevor ein in einer Hochsprache geschriebenes Programm ausgeführt werden kann. Diese Aufgabe wird von einem Compiler ausgeführt, in diesem Fall dem Borland C++ Compiler (siehe 4.1). Die abstrakte Sprache ist plattformunabhängig, lediglich ein für die Plattform geschriebener Compiler muss verwendet werden.

C gilt als imperative und strukturierte Programmiersprache.

Imperativ bedeutet, dass das Vorgehen durch eindeutige Anweisungen festgelegt wird und nicht, wie bei deklarativen Programmiersprachen, nur was berechnet werden soll und nicht wie dies zu geschehen hat.

Eine Programmiersprache wird als strukturiert bezeichnet, wenn eine Zerlegung in einzelne Prozeduren möglich ist und ein komplettes Programm lediglich aus Verzweigungen, Schleifen besteht und Programmanweisungen hintereinander ausgeführt werden.

2.3 Übertragungsprotokolle

2.3.1 Ethernet 10/100Base-T Halbduplex

Der Begriff *Ethernet* bezeichnet eine Gruppe von Netzwerk-Übertragungstechnologien, die hauptsächlich für lokale Datennetze Verwendung findet. Im OSI-Referenzmodell ist Ethernet auf Ebene 1 angesiedelt, der physikalischen Schicht.

Ethernet ist durch den Standard IEEE 802.3 festgelegt, die einzelnen Untertypen unterscheiden sich unter anderem in der erreichbaren Geschwindigkeit. Die Daten werden in sogenannten Datenframes versendet, die aus einem Header, dem Datenteil und einem Trailer bestehen. Der Header wiederum setzt sich zusammen aus den MAC-Adresse von Sender und Empfänger sowie Steuerinformationen zur Datenflusskontrolle. Der Trailer ist eine Prüfsumme, die verwendet wird, um Übertragungsfehler festzustellen und zu korrigieren, soweit dies möglich ist.

Der hier verwendete Mikrocontroller unterstützt *10/100Base-T Ethernet* im *Halbduplex*-Modus. Die 10 bzw die 100 geben die maximal erreichbare Geschwindigkeit an, wobei 10 für 10 Mbit/s steht und 100 für 100 Mbit/s. Der Begriff *Base-T* zeigt an, dass ein Kabel mit verdrehten Adernpaaren verwendet wird, ein sogenanntes *Twisted-Pair*-Kabel.

Halbduplex gibt die Betriebsart der Kommunikation an. Bei Halbduplex können Nachrichten über ein Kabel in beide Richtungen ausgetauscht werden, aber nicht zeitgleich. Wenn ein Client gerade als Sender fungiert, kann kein weiterer Client senden sondern muss warten, bis der Sendevorgang beendet ist. Ein Beispiel für dieses Verfahren sind Walkie-Talkies, bei denen man nur senden kann, wenn man einen Knopf drückt und nur empfangen kann, wenn man diesen nicht drückt.

Zur Regelung, des Sendevorgangs wird bei Ethernet der *CSMA/CD*-Algorithmus verwendet. Dieser sieht vor, dass jeder der Clients auf der Leitung horcht, ob gerade gesendet wird und nur tatsächlich sendet, wenn diese frei ist. Sollten einmal mehrere Clients gleichzeitig anfangen zu S-enden, so wird dies detektiert und alle brechen den Sendevorgang ab. Nach einer zufälligen Zeit beginnt das Prozedere dann von vorne. Durch die zufällige Wartezeit ist es wahrscheinlich, dass beim folgenden Versuch alle ohne Probleme senden können. Mit diesem Algorithmus werden höhere Geschwindigkeiten erzielt als beispielsweise mit *Token Passing*, bei dem der Reihe nach jeder Client nur zu einer bestimmten Zeit senden darf. [20]

2.3.2 Serial Peripheral Interface

Das Serial Peripheral Interface, kurz SPI, dient hauptsächlich der seriellen Kommunikation zwischen einem Hostprozessor, *Master* genannt, und Peripheriebausteinen, die als *Slave* bezeichnet werden.

Die Datenübertragung wird über zwei Datenleitungen durchgeführt, über MOSI und MISO. MOSI steht für *Master Out Slave In*, ist also die Datenleitung, mit der der Master Daten an den Slave sendet. MISO bedeutet *Master In Slave Out* und wird für die umgekehrte Richtung verwendet. Ein anderer Begriff für MISO ist SDO (*Serial Data Out*), einer für MOSI ist SDI (*Serial Data In*).

Über einen SPI-Bus können mehrere Slaves angesprochen werden, die im Regelfall parallel geschaltet sind. Mittels einer Chip-Select-Leitung, kurz *CS*, wird bestimmt, welcher Slave die vom Master gesendeten Bits interpretieren soll. Für jeden Slave ist normalerweise eine CS-Leitung nötig, die auf Ground gezogen wird, wenn der jeweilige Chip aktiv werden soll. In besonderen Fällen, in denen die Slaves in Reihe geschaltet sind, kann es auch sein, dass mehrere Slaves sich eine CS-Leitung teilen. Eine weitere geläufige Bezeichnung für die CS-Leitung ist *Slave-Select*, kurz *SS*.

Die Datenübertragung funktioniert wie bei einem Schieberegister. Damit dies fehlerfrei abläuft, müssen Master und Slave im selben Takt lesen bzw. schreiben, weswegen eine vierte Leitung zwischen den beiden notwendig ist. Diese Leitung wird als SCK bezeichnet, als *Serial Clock*. SCK und CS sind Steuerleitungen. [27]

Der Aufbau eines SPI-Interfaces wird in Abbildung 4 anhand eines Beispiels mit drei Slaves veranschaulicht.

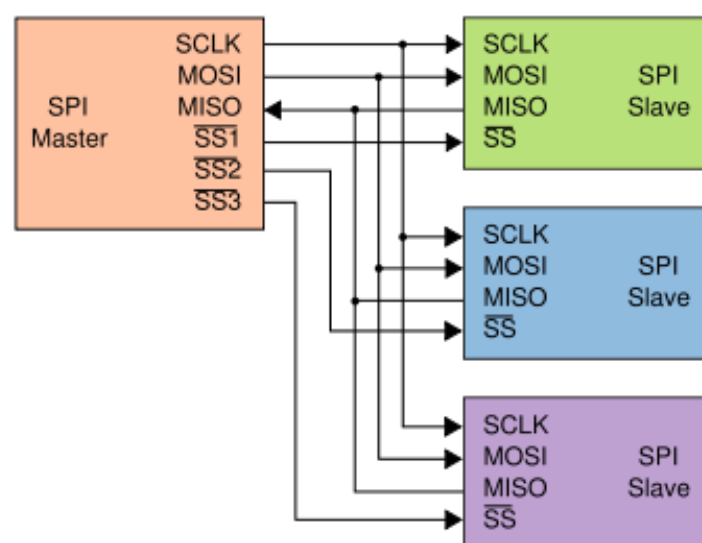


Abbildung 4: Sternförmige SPI-Busstruktur mit drei Slaves [11]

3 Verwendete Hardware

3.1 Mikrocontroller *SC13* von Beck-IPC

Zur Steuerung des Datensimulators wird ein Mikrocontroller vom Typ SC13 des Herstellers Beck-IPC verwendet. Darin ist ein Intel 80186-kompatibler Prozessor mit 40 MHz verbaut, der aufgrund seines Aufbaus besonders für die Verwendung in Mikrocontrollern geeignet ist. Unter anderem zeichnet er sich durch einen geringen Stromverbrauch (300 mA) und eine niedrige benötigte Versorgungsspannung (5 V) aus.

Neben dem Prozessor sind in dem Mikrocontroller 512 kB Arbeitsspeicher sowie 512 kB Flashspeicher eingebaut. Im Flashspeicher befindet sich das *Real-Time Operating System*, kurz RTOS, also das Betriebssystem. Der restliche Flashspeicher steht für die Applikationen des Anwenders zur Verfügung. Zusätzlich kann der verfügbare Speicher durch eine CompactFlash-Speicherkarte erweitert werden.

Der SC13 kann mehrere Schnittstellen ansteuern. Neben einer 10/100BaseT-Ethernet-Schnittstelle kann er auch über zwei serielle Schnittstellen des Typs RS-232 angesprochen werden. Weiterhin existieren ein externer 8-Bit-Daten/Adress-Bus sowie 6 Eingänge für externe Interrupts und 14 programmierbare Input-/Output-Pins, kurz PIO-Pins. Mittels der PIO-Pins kann zusätzlich ein SPI- und ein I²C-Bus simuliert werden.

Jedoch ist zu beachten, dass der Mikrocontroller insgesamt nur über 32 Pins verfügt und deswegen eine Mehrfachbelegung der Pins nötig ist. So ist beispielsweise bei Nutzung der seriellen Schnittstelle S1 die Verwendung der programmierbaren Input-/Output-Pins PIO5, PIO6, PIO11 und PIO12 sowie der externen Interrupts INT2, INT3 und INT4 nicht möglich, da sie die gleichen Pins verwenden wie die serielle Schnittstelle. [1] [2]



Abbildung 5: Mikrocontroller SC13 [2]

Wie bereits erwähnt, verfügt der SC13 über ein RTOS, also ein Echtzeitbetriebssystem. In das Betriebssystem sind unter anderem ein FTP-Server, ein Web-Server, ein Telnet-

Server, ein DHCP-Client sowie ein DNS-Client integriert. Zusätzlich hat das Betriebssystem ein Dateisystem, so dass die Entwicklung von Anwendungen möglich ist, die vom Betriebssystem unabhängig sind. Weiterhin können dadurch Websites in gewohnter Art und Weise in HTML erstellt werden.

Ein lesender und schreibender Zugriff auf das Dateisystem ist über den FTP-Server möglich. Ein Telnet-Client ermöglicht das Ausführen von Programmen auf dem Mikrocontroller sowie den Zugriff auf Funktionen des Betriebssystems. Das Betriebssystem unterstützt bis zu zwölf aktive Applikationen parallel. [6]

3.2 Entwicklungsboard *DK50* von Beck-IPC

Das Entwicklungsboard DK50 ist eine Platine, die alle Grundfunktionen des Mikrocontrollers SC13 unterstützt.

Neben dem Sockel für den SC13 sind verschiedene externe Schnittstellen, Status- und Traffic-LEDs sowie die Spannungsversorgung und ein Reset-Schalter auf dem Entwicklungsboard vorhanden. Die externen Schnittstellen sind ein Ethernet-Anschluss, zwei serielle Schnittstellen, eine Halterung für CompactFlash-Karten und je acht digitale Input- bzw. Output-Pins. Diese elektronischen Bauteile sind durch Leiterbahnen an den Mikrocontroller angeschlossen. Dennoch ist es möglich, sämtliche Pins des Mikrocontrollers direkt zu erreichen, da diese an der Unterseite des Boards abgegriffen werden können. Der Aufbau des Entwicklungsboards wird durch das Blockschaltbild im Anhang verdeutlicht (S. F).

Im Anhang befinden sich zum Entwicklungsboard eine beschriftete Zeichnung (S. E und ein Schaltplan (S. G).

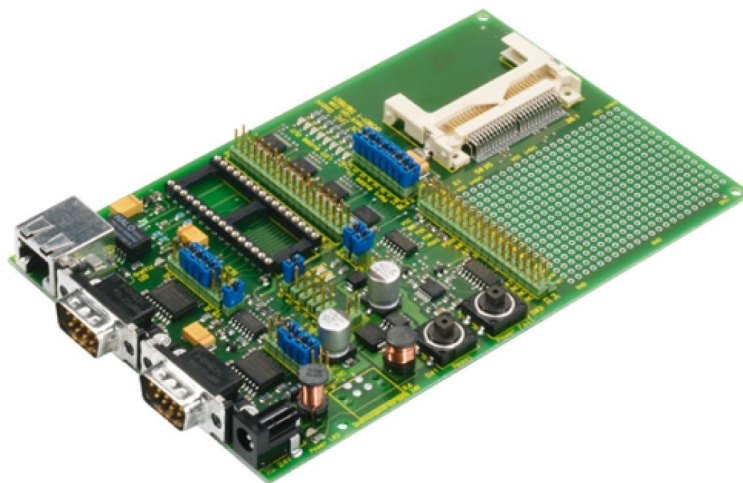


Abbildung 6: Entwicklungsboard DK50 ohne Mikrocontroller [3]

3.3 Drucksensor *SCP1000-D01* von VTI

Der SCP1000-D01 der Firma VTI ist ein Druck- und Temperatursensor. In diesem Projekt wird der Sensor verwendet, um eben diese Werte zu ermitteln und an den Mikrocontroller zu senden.

Die Produktfamilie SCP1000 gibt es in zwei Ausführungen, die sich lediglich durch die Schnittstelle unterscheiden: Der verwendete SCP1000-D01 verfügt über einen SPI-Bus, der -D11 über einen I²C-Bus.

Der Messbereich der Druckmessung liegt zwischen 300 und 1200 hPa, der der Temperaturmessung zwischen -20 und +70 °C. Die Betriebsspannung sollte zwischen +2,4 und +3,3 V liegen.

Die absoluten Grenzdaten, deren Über- bzw. Unterschreitung zu einer Schädigung des Chips führen kann, sind für die Temperatur -30 und +85 °C. Da im Rotorversuchstand, dem späteren Verwendungsort des Sensors, mit etwa +20 °C und 1000 hPa Normbedingungen herrschen, ist diese Spanne vollkommen ausreichend. Die zulässige Versorgungsspannung ist -0,3 und +3,6 V. Es existiert also ein Sicherheitsspielraum von 10 bis 15 °C und 0,3 V. Die Spannung an den Input- und Output-Pins muss mindestens -0,3 V betragen und darf höchstens 0,3 V über der Versorgungsspannung liegen. Diese Informationen gehen aus [29] hervor.

Der SCP1000 verfügt über 18 SMD-Kontakte, liegt für dieses Projekt allerdings bereits auf einem PCB-Carrier vor, so dass eigenständiges Löten ohne größere Schwierigkeiten möglich ist.



Abbildung 7: Druck- und Temperatursensor SCP1000 von VTI in Detailansicht (links) [29] und auf dem Test PCB Carrier (rechts) [31]

Der Sensor stellt vier verschiedene Messmodi zur Verfügung, der *High Resolution Mode*, der *High Speed Mode*, der *Ultra Low Power Mode* sowie der *Low Power Mode*. Die ersten drei bieten eine kontinuierliche Messung, wohingegen im *Low Power Mode* eine Messung durch einen externen Trigger ausgelöst wird. In diesem Modus kann gewählt werden zwischen der Auflösung des Drucks von 17 bit und 15 bit. Allerdings ist an die Auflösung die maximale Triggerfrequenz gekoppelt: Beträgt die Auflösung 15 bit, so ist die maximale Frequenz 9 Hz. Wird jedoch die höhere Auflösung gewählt, kann nur maximal mit 1.8 Hz gemessen werden. Unabhängig davon, mit welcher Frequenz gemessen wird oder wie

hoch die Auflösung des Luftdrucks ist, beträgt die Auflösung der Temperatur stets 14 bit. Der Stromverbrauch ist abhängig von der Auflösung und der Triggerfrequenz, liegt jedoch stets zwischen $1\ \mu\text{A}$ und $25\ \mu\text{A}$.

Im *High Resolution Mode* beträgt die Auflösung der Temperatur ebenfalls 14 bit, die der Temperatur ist unveränderbar 17 bit und die Aktualisierungsrate 1.8 Hz. Der Stromverbrauch liegt bei $25\ \mu\text{A}$.

Eine Auflistung der genauen Spezifikationen aller Messmodi ist in Tabelle 1 aufgeführt.

Tabelle 1: Messmodi des Sensors SCP1000 [30]

Mode	Luftdruck [bit]	Temperatur [bit]	Aktualisierungsrate [Hz]	Stromverbrauch [μA]
High Resolution	17	14	1.8	25
High Speed	15	14	9	25
Ultra Low Power	15	14	1	3.5
Low Power Mode	17	14	max. 1.8	1-25
Low Power Mode	15	14	max. 9	1-25

3.4 Display *GU128x18F-K612A4* von Noritake Itron

Das verwendete Display ist das GU128x18F-K612A4 des Herstellers Noritake Itron. Es handelt sich um ein *Punkt-Matrix VFD* Display.

Punkt-Matrix-Display bedeutet, dass die Anzeige aus horizontal und vertikal angeordneten Punkten besteht, die mittels der Zeilen- und Spaltennummer eindeutig festgelegt und einzeln ansprechbar sind. Das vorliegende Display verfügt über 128 Bildpunkte in der Horizontalen und 18 Bildpunkte in der Vertikalen.

VFD zeigt an, dass es sich um eine Vakuum-Fluoreszenzanzeige handelt. Diese basiert darauf, dass die verwendeten Materialien fluoreszieren, wenn sie von Elektronen getroffen werden, die ein über dem Grenzwert liegendes Energieniveau aufweisen.

Das GU128X18F verfügt über zwei serielle Schnittstellen, zum einen einen SPI-Bus und zum anderen eine bzw. RS-232. Zusätzlich bietet das Display acht I/O-Pins, über die beispielsweise Tasten angeschlossen werden können.

Das Display stellt drei verschiedene Schriftarten zur Verfügung. Zwischen einer proportionalen Schriftart und zwei Monospace-Schriftarten kann gewählt werden. Die Zeichen der Monospace-Schriftarten sind 5x7 bzw. 10x14 Punkte groß, bei den Zeichen der proportionalen Schriftart variiert dies, das Maximum ist jedoch 5x5.

Mit der Wahl der Schriftart wird also auch die Anzahl der auf dem Display darstellbaren Textzeilen bestimmt. Da das Display 18 Bildpunkte hoch ist, können mit der proportionalen Schriftart drei Zeilen und mit der kleinen Monospace-Schriftart zwei Zeilen dargestellt

werden. Mit der großen Monospace-Schriftart ist lediglich die Anzeige einer einzigen Textzeile möglich.

Neben der direkten Eingabe von Buchstaben ist es auch möglich, Grafiken darzustellen. Hierbei werden die Bitmuster der übertragenen Bytes auf die Punkt-Matrix des Displays gelegt. An jeder Stelle, an der das Bitmuster eine '1' enthält, wird der Punkt auf dem Display angesprochen.

Ebenfalls möglich ist es, bestimmte Bereiche des Displays komplett anzuwählen oder zu invertieren.

Auch bietet das Display die Möglichkeit, bestimmte Display-Konfigurationen als Makro zu speichern, beispielsweise per Bitmuster erzeugte Grafiken. Anschließend kann die komplette Konfiguration mit einem einzigen Befehl aufgerufen werden.



Abbildung 8: Display GU128x18F-K612A4 [19]

4 Verwendete Software

4.1 Borland C++ und Borland C++ Compiler

Borland C++ ist eine integrierte Entwicklungsumgebung, die die Programmiersprachen C und C++ unterstützt. Die Software-Entwicklung wird beispielsweise durch einen Debugger unterstützt, sodass während der Entwicklung das Programm schrittweise ausgeführt werden kann, was die Fehlersuche deutlich erleichtert.

Als Compiler wird der Borland C++ Compiler verwendet, ein Cross-Compiler. Er kann also so konfiguriert werden, dass das erstellte Programm auf einer anderen Prozessorarchitektur und/oder auf einem anderen Betriebssystem läuft. Cross-Compiler werden in der Regel aus Performance-Gründen verwendet. Vorallem bei Mikrocontrollern ist es aufgrund zu geringer Rechenleistung einfach nicht möglich, den Programm-Code direkt auf der Zielplattform zu kompilieren. Aber auch wenn dies möglich ist, wird teilweise ein Cross-Compiler auf einem leistungstärkeren System verwendet, um Zeit zu sparen.

4.2 CHIPtool

CHIPtool von Beck wird in der Version 5.11.1.0 verwendet. Die Software erlaubt es, den Mikrocontroller über das Netzwerk zu konfigurieren, also beispielsweise die IP-Adresse und die Netzmaske zuzuweisen. Ebenfalls sind ein FTP- und ein Telnet-Client integriert. Mit dem FTP-Client ist ein Zugriff auf das Dateisystem des Mikrocontrollers möglich, d.h. er kann unter Anderem dazu verwendet werden, das kompilierte Programm zu überspielen oder die Konfigurationsdatei des Mikrocontrollers zu modifizieren.

Mittels des Telnet-Protokolls ist ein Zugriff auf Kommandozeilen-Ebene auf das RTOS des Mikrocontroller möglich, so können zum Beispiel Programme gestartet oder Konfigurationen eingesehen werden.

Weiterhin ist es mit CHIPtool möglich, eine neue Version des Betriebssystems aufzuspielen oder ein Software-Image für die Serienproduktion zu erstellen. Diese Funktionen wurden allerdings nicht im Verlauf des Projekts nicht genutzt. [9]

4.3 Mozilla Firefox

Der freie Webbrowser Mozilla Firefox dient dazu, Websites aufzurufen. In diesem Projekt findet er Verwendung beim Zugriff auf die Weboberfläche des Mikrocontrollers.

Anstelle des DLR-Standards, dem Microsoft Internet Explorer, wird Mozilla Firefox verwendet, da im späteren Betrieb mit diesem auf die Weboberfläche zugegriffen werden wird.

4.4 Institutseigene Software

Als Institutseigene Software werden Anwendungen bezeichnet, die von Mitarbeitern des DLR für andere Projekte erstellt und in diesem Projekt eingesetzt wurde.

4.4.1 Extserver

Bei Tests im Rotorversuchstand dient das Programm Extserver als Zwischenspeicher und Verteiler von asynchronen Daten, die von verschiedenen Quellen mittels des TCP/IP-Protokolls als ASCII-String gesehendet werden. Die Quellen der Daten werden mittels der IP-Adresse ermittelt, in einer Datenbank ist die Namenskennung hinterlegt.

Clients haben die Möglichkeit, aktuelle oder aufgezeichnete Datensätze per TCP/IP anzufordern.

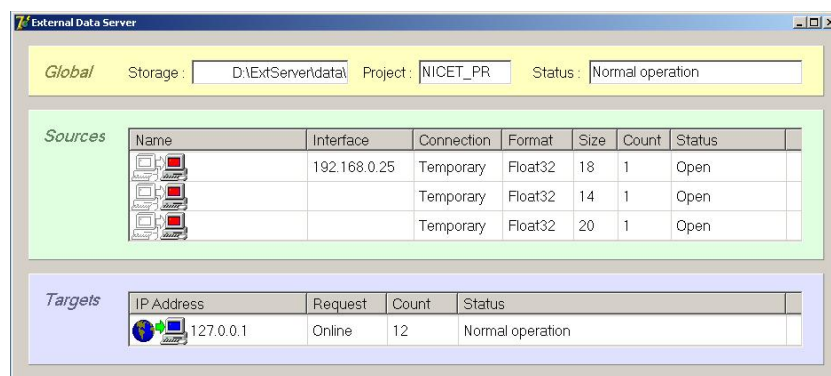


Abbildung 9: Institutseigene Software Extserver

4.4.2 Quickview

Die Anwendungssoftware Quickview ist einer der in Abschnitt 4.4.1 erwähnten Clients. Um die aktuellen Datensätze in Echtzeit auszulesen wird eine Anfrage an das Programm Extserver übermittelt, woraufhin dieser die Daten sendet. Es kann unter Anderem ausgewählt werden, welche Informationen angezeigt und wie oft diese aktualisiert werden sollen.

Die Verwendung von Extserver und Quickview ermöglichen es, die Funktionalität der Netzwerkkomponente des erstellten Programms zu testen, ohne direkten Zugang zu dem Netzwerk im Rotorversuchstand zu verfügen, in dem die Anwendung später laufen wird.

5 Anbindung der Peripherie-Geräte

5.1 Ethernet

5.2 Stromversorgung

5.3 Luftfeuchtesensor

Zunächst wurde überprüft, inwieweit die Luftfeuchte Einfluss auf die weiteren Ergebnisse hat. Dies werde ich exemplarisch an der Schallgeschwindigkeit zeigen.

Die Schallgeschwindigkeit c berechnet sich aus

$$c = \sqrt{\kappa \frac{p}{\rho}} \quad (1)$$

wobei κ der Isotropenexponent ist, ρ die Dichte der Luft und p der Luftdruck. Der Isotropenexponent ist abhängig von der Struktur der enthaltenen Moleküle, also auch abhängig von dem Anteil an Wassermolekülen in der Luft, was gleichbedeutend mit der Luftfeuchte ist.

Der Isotropenexponent eines Gases berechnet sich aus

$$\kappa = \frac{C_p}{C_V} \quad (2)$$

wobei C_p und C_V molare Wärmekapazitäten sind, auf die im folgenden noch näher eingegangen wird.

Da Luft ein Gemisch aus verschiedenen Gasen ist, berechnet sich der Isotropenexponent der Luft aus den anteiligen Isotropenexponenten der Bestandteile.

$$\kappa_{Luft} = \sum (\kappa_{Bestandteil} * Volumenanteil) = \sum \left(\frac{C_{p_{Bestandteil}}}{C_{V_{Bestandteil}}} * Volumenanteil \right) \quad (3)$$

C_p ist die molare Wärmekapazität eines Gases bei konstantem Druck, C_V die bei konstantem Volumen. Laut [25] steht C_V bei idealen Gasen in direktem Zusammenhang mit der Anzahl der in dem Molekül enthaltenen Atomen und der Struktur des Moleküls. Die Werte sind in Tabelle 2 zusammengefasst.

C_p und C_V stehen in dem Verhältnis

$$C_p - C_V = R \quad (4)$$

zueinander, wodurch sich die in Tabelle 3 dargestellten Werte für C_p und C_V ergeben.

Setzt man nun die Werte der in Tabelle 4 dargestellten Zusammensetzung trockener Luft und Isotropenexponenten der einzelnen Luftbestandteile in Gleichung 3 ein, erhält man den Isotropenexponenten von trockener Luft.

Tabelle 2: molare Wärmekapazität bei konstanten Volumen mit der Gaskonstante R

Molekültyp	C_V
1-atomige Gase	$\frac{3}{2}R$
2-atomige Gase	$\frac{5}{2}R$
3- oder mehr-atomige gestreckte Gase	$\frac{5}{2}R$
3- oder mehr-atomige gewinkelte Gase	$3R$

Tabelle 3: molare Wärmekapazität mit der Gaskonstante R

Molekültyp	C_V	C_P
1-atomige Gase	$\frac{3}{2}R$	$\frac{5}{2}R$
2-atomige Gase	$\frac{5}{2}R$	$\frac{7}{2}R$
3- oder mehr-atomige gestreckte Gase	$\frac{5}{2}R$	$\frac{7}{2}R$
3- oder mehr-atomige gewinkelte Gase	$3R$	$4R$

$$\kappa_{0\%Luftfeuchte} = 1.40152 \quad (5)$$

Tabelle 4: Zusammensetzung trockener Luft

Gas	Strukturformel	Struktur	$\kappa_{Bestandteil}$	Volumenanteil [%]
Stickstoff	N_2	gestreckt	$\frac{7}{5}$	78.1
Sauerstoff	O_2	gestreckt	$\frac{7}{5}$	20.9
Argon	Ar	gestreckt	$\frac{5}{3}$	0.9
Kohlenstoffdioxid	CO_2	gewinkelt	$\frac{4}{3}$	0.039

Luftfeuchte wird in Prozent gegenüber der maximalen Sättigung angegeben. Bei tropischen Bedingungen beträgt die maximale Sättigung und damit eine Luftfeuchte von 100% von Wasserdampf in der Luft 4 Volumenprozent [28]. Das Gewicht eines Kubikmeter Luft beträgt nach eigenen Messungen durchschnittlich 1.18 kg. Wird nun der Wasserdampf-Anteil, der aus 3-atomigen gewinkelten Molekülen besteht, anteilig in die Berechnung des Isotropenexponenten mit einbezogen ergibt sich

$$\kappa_{100\%Luftfeuchte} = 1.39879 \quad (6)$$

Der absolute Unterschied von κ zwischen 0% Luftfeuchte und 100% beträgt also 0.00273.

Mit durchschnittlichen Werten, Luftdruck ist 101325 Pa und Luftdichte ist $1.18 \frac{kg}{m^3}$ ergeben sich für $\kappa_{0\%Luftfeuchte}$ und $\kappa_{100\%Luftfeuchte}$ folgende Rechnungen:

$$\kappa_{0\%Luftfeuchte} = \sqrt{\kappa_{0\%Luftfeuchte} \frac{101325 Nm^{-2}}{1.18 kg m^{-3}}} = 346.9101066 \frac{m}{s} \quad (7)$$

$$c_{100\%Luftfeuchte} = \sqrt{\kappa_{100\%Luftfeuchte} \frac{101325 Nm^{-2}}{1.18 kgm^{-3}}} = 346.5723852 \frac{m}{s} \quad (8)$$

Der absolute Fehler, der sich aus den Ergebnissen der Rechnungen 7 und 8 ergibt ist $F = 0.337721421$. Daraus ergibt sich ein relativer Fehler von $f = 0.0974\%$.

Selbst der relative Fehler von 0.0974% ergibt sich nur, wenn von 0% Luftfeuchte ausgegangen wird, tatsächlich aber 100% herrschen. Hinzu kommt, das von tropischen Bedingungen ausgegangen wurde, die im Rotorversuchstand, wenn überhaupt, nur im Hochsommer zustande kommen. Wird stattdessen von einem mittleren Wert für die Luftfeuchte und einer Temperatur von $20^\circ C$ ausgegangen, ist die maximale Abweichung vom tatsächlichen Wert noch weit geringer, wodurch der relative Fehler weiter abnimmt.

Aufgrund des zu geringen Einflusses der Luftfeuchte auf die relevanten Ergebnisse wurde beschlossen, keinen Luftfeuchtesensor zu verwenden. Der Isotropenexponent wurde auf den Wert 1.4 fest eingestellt, was einer Luftfeuchte von

5.4 Druck- und Temperatursensor

Als Druck- und Temperatursensor wurde der SCP1000 von VTI verwendet. Eine genauere Beschreibung zu diesem Sensor ist in Abschnitt 3.3 zu finden.

5.4.1 Aufbau der Hardware

Der Drucksensor SCP1000 wird über ein SPI-Interface mit dem Mikrocontroller verbunden.

5.4.2 Entwicklung der Software

5.5 Display

5.5.1 Aufbau der Hardware

5.5.2 Entwicklung der Software

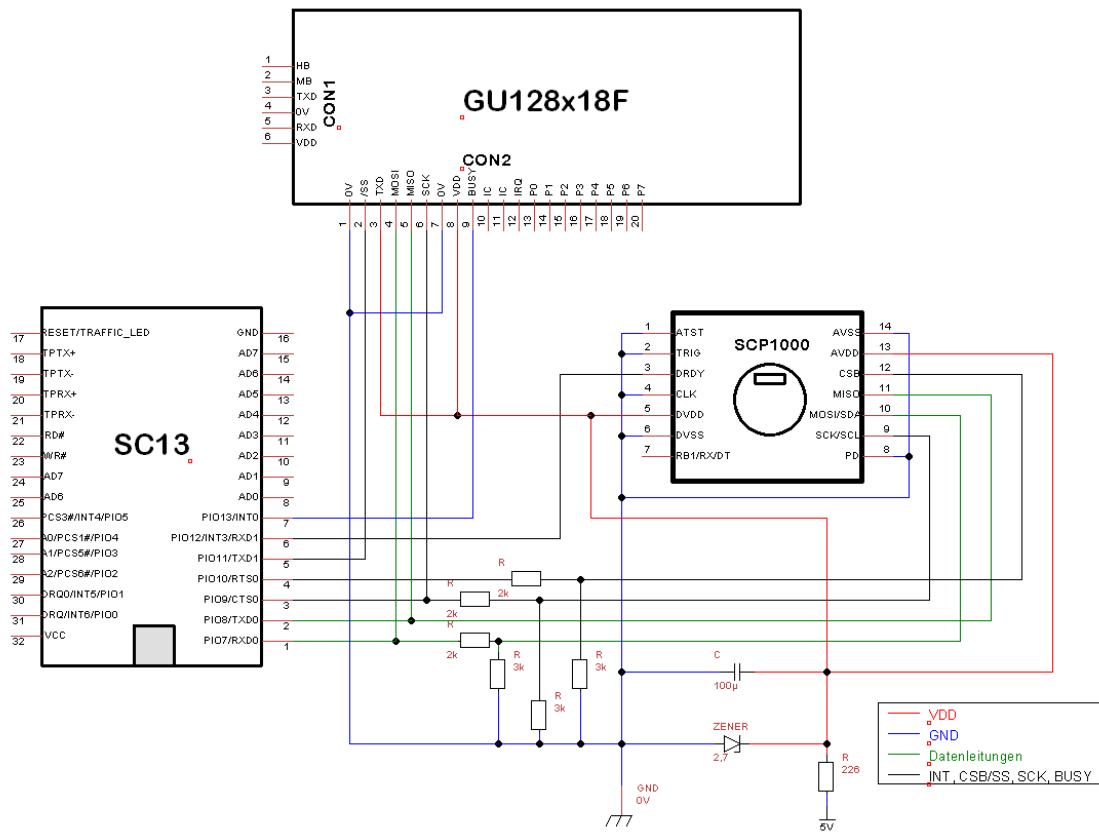


Abbildung 12: Schaltbild mit Mikrocontroller SC13, Display und Sensor SCP1000

6 Tests

6.1 Funktionstest

6.2 Dauertest

7 Ergebnisse der Arbeit

8 Fazit

Literaturverzeichnis

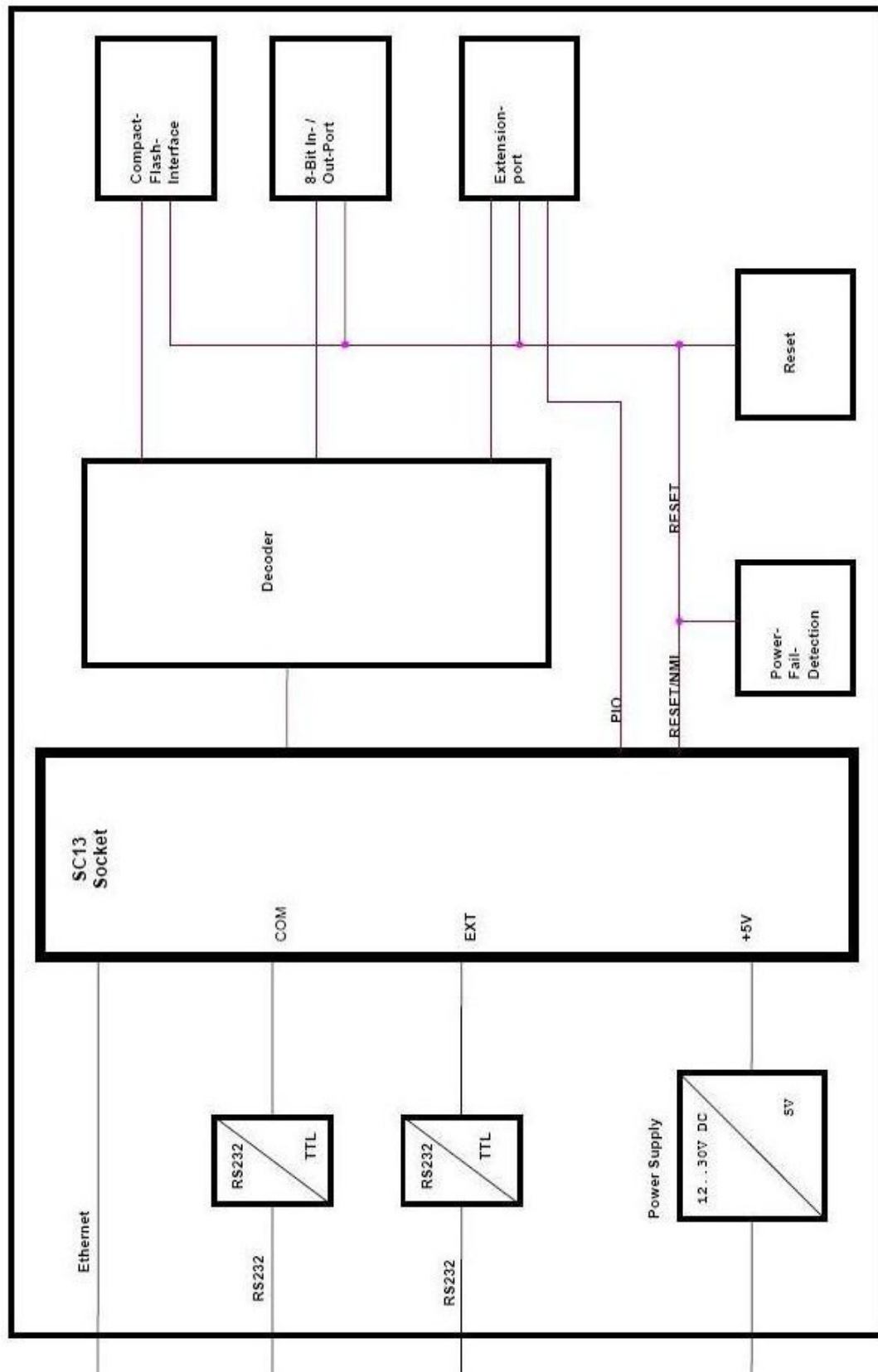
- [1] BECK IPC: *SC13 Datenblatt*.
http://www.beck-ipc.com/files/datasheet/DS_SC13_V5.pdf, 05.01.2010
- [2] BECK IPC: *SC13 Datenblatt*.
http://www.beck-ipc.com/files/manual/SC1XHW_v19_080801.pdf, 07.01.2010
- [3] BECK IPC: *DK50 Hardware Handbuch Version 1.5 - Build 20.03.2006*.
http://www.beck-ipc.com/files/manual/dk50_manual_v15_20032006.pdf,
18.01.2010
- [4] BECK IPC: *DK50 Schaltplan V 1.2 - H02*.
http://www.beck-ipc.com/files/diagram/cd_dk50_v12.pdf, 18.01.2010
- [5] BECK IPC: *DK50 Datenblatt Version 3.0*.
http://www.beck-ipc.com/files/datasheet/ds_dk50_v3.pdf, 20.01.2010
- [6] BECK IPC: *IPC@CHIP RTOS Documentation [Build 28.09.2009]*.
http://www.beck-ipc.com/files/manual/ipcatchip_rtos_documentation_build_090928.pdf, 20.01.2010
- [7] BECK IPC: *Leitfaden für Hardwareentwurf mit SC1x*.
http://www.beck-ipc.com/files/applicationnotes/AN_SC1x-DesignGuide.pdf, 20.01.2010
- [8] BECK IPC: *SC12/SC13 Ethernet*.
http://www.beck-ipc.com/files/applicationnotes/an_ethernet_sc12_sc13_v4.pdf, 20.01.2010
- [9] BECK IPC: *IPC@Chip - Getting Started*.
http://www.beck-ipc.com/de/download/load.asp?f=/manual/IPCatCHIP_Gettingstart_V24.pdf, 22.06.2010
- [10] BECK IPC: *RTOS - Real-Time Operation System*.
<http://www.beck-ipc.com/de/products/rtos/index.asp>, 24.02.2010
- [11] BURNETT, COLIN M.L.: *Serial Peripheral Interface*.
http://upload.wikimedia.org/wikipedia/commons/f/fc/SPI_three_slaves.svg, 15.03.2010
- [12] DLR: *Institut für Flugsystemtechnik - Forschungsarbeit*.
http://www.dlr.de/ft/desktopdefault.aspx/tabid-1359/1862_read-3347/,
08.03.2010

-
- [13] DLR: *Institut für Flugsystemtechnik - Rotorversuchstand.*
http://www.dlr.de/ft/desktopdefault.aspx/tabid-1387/1915_read-3372/,
08.03.2010
- [14] DLR: *DLR testet neue Hubschraubersteuerung: Erfolgreicher Erstflug mit zwei aktiven Sidesticks.*
http://www.dlr.de/DesktopDefault.aspx/tabid-1/86_read-20504/gallery-1/gallery_read-Image.1.5427/, 09.03.2010
- [15] DLR: *Das DLR im Überblick.*
http://www.dlr.de/desktopdefault.aspx/tabid-636/1065_read-1465/,
19.01.2010
- [16] DNW: *German-Dutch Wind Tunnels: LLF.*
<http://www.dnw.aero/windtunnels.aspx?id=278&menuid=34&subid=278>,
09.03.2010
- [17] EARTH: *MUM-T Sicherheitstechnologien für den gemeinsamen Einsatz bemannter und unbemannter Luftfahrzeuge.*
http://www.dlr.de/ft/desktopdefault.aspx/tabid-1364/1937_read-17452/,
10.03.2010
- [18] ESRL GLOBAL MONITORING DIVISION: *Carbon Cycle Toolkit.*
http://www.esrl.noaa.gov/gmd/education/carbon_toolkit/basics.html,
23.06.2010
- [19] FARNELL: *Noritake - GU128X18F-K612A4 - VFD Modul 128X18 Graphik.*
<http://de.farnell.com/noritake-itron/gu128x18f-k612a4/vfd-modul-128x18-graphik/dp/1216679>, 04.03.2010
- [20] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS: *IEEE 802.*
<http://www.ieeeusa.org/policy/policy/2001/01aug27IEEE802.pdf>, 16.03.2010
- [21] KÖNIG, Christian: *Bericht zum Modul Praxis I - Thema 1: Netzwerkbasierter Datensimulator für den Rotorversuchstand.* DLR, 2009
- [22] NORITAKE ITRON: *Dot Graphic VFD Module - GU128X18F-K612A4.*
<http://www.noritake-itron.com/Specs/GUK/GU128x18F-K612A4-03.pdf>,
05.03.2010
- [23] PRINZ, Peter ; KIRCH, Ulla: *C. Einführung und professionelle Anwendung. IT-Studienausgabe.*
2., aktualisierte Auflage. REDLINE GMBH, 2007
- [24] RITCHIE, Dennis M. ; KERNIGHAN, Brian W.: *Programmieren in C.*
2. Auflage. Carl Hanser Verlag München, 1990
-

- [25] RÖMPP ONLINE: *Molwärme*.
<http://www.roempp.com/prod/?ID=RD-13-02959>, 24.06.2010
- [26] SCHWARZ, ANDREAS: *Mikrocontroller*.
<http://www.mikrocontroller.net/articles/Mikrocontroller>, 11.03.2010
- [27] SCHWERDTFEGER, MARTIN: *SPI - Serial Peripheral Interface*.
<http://www.mct.de/faq/spi.html>, 15.03.2010
- [28] UNI KOBLENZ: *Relative und absolute Luftfeuchtigkeit*.
<http://www.uni-koblenz.de/~odsbcg/rhnebel/relative.htm>, 24.06.2010
- [29] VTI TECHNOLOGIES: *SCP1000 Data Sheet*.
http://www.vti.fi/midcom-serveattachmentguid-9cbae6a382efd245cb62354a54ff62c7/scp1000-d01_-d11_pressure_sensor_datasheet_28-08-2007.pdf, 20.01.2010
- [30] VTI TECHNOLOGIES: *SCP1000 Product Family*.
http://www.vti.fi/midcom-serveattachmentguid-e9735f5baea5692c3b9753f9069511c7/scp1000_product_family_specification_rev_0.08.pdf, 20.01.2010
- [31] VTI TECHNOLOGIES: *SCP1000 Test PCB Carrier*.
http://www.vti.fi/midcom-serveattachmentguid-a5eab1b6104f11dd83cf7774ee6a7f057f05tn38_scp1000_test_pcb_carrier_rev_1.3.pdf, 23.02.2010

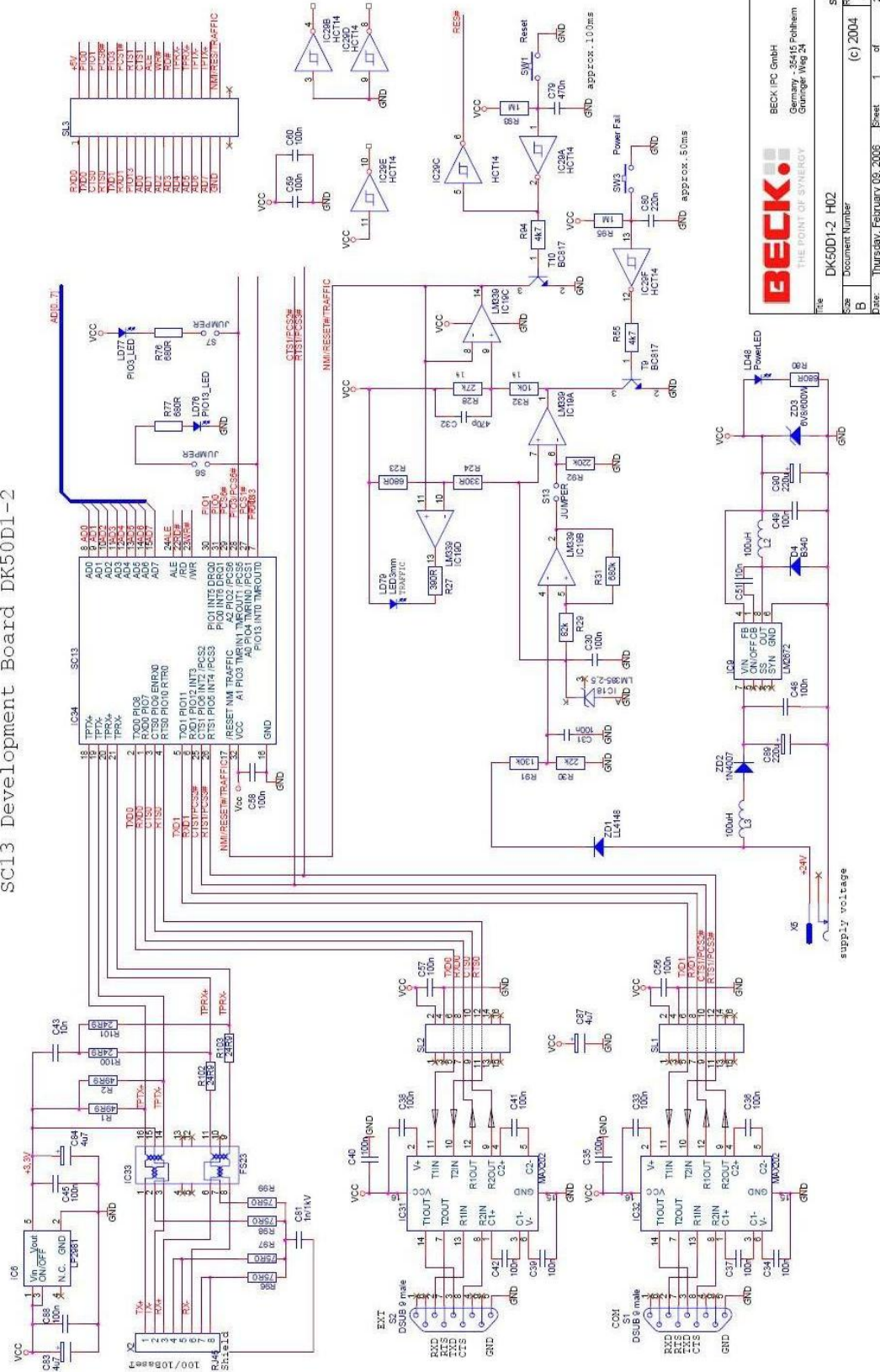
Anhang

Blockschaltbild des DK50 [3]

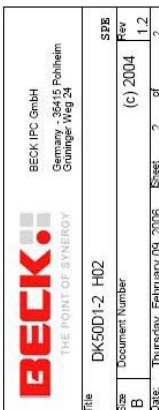


Schaltplan des DK50 [3]

SC13 Development Board DK50D1-2



BECK THE POINT OF SYNERGY		BECK PC GmbH Germany - 35415 Pöhlheim Grunger Weg 24	
Title	DK50D1-2 H02	SPB	
Doc. Number		(c) 2004	
Rev		1.2	
Date	Thursday, February 09, 2006	Sheet	1 of 2



Datenblatt des SC13 [1]

Datasheet: SC13-LF (RoHS compliant)

Mature Product - Not recommended for new designs

Embedded Web-Controller

SC186-CPU @40 MHz

512KB RAM

512KB Flash

10/100BaseT Ethernet

Software runtime licence SL01 included



Technical data

Processor	SC186/40 MHz (AM186 compatible)
Working Memory (RAM)	512KB (free user space see API documentation)
Flash Memory (RTOS and Flash disk)	512KB (free user space see API documentation)
Serial	2xTTL port (RxD, TxD, RTS, CTS with DMA)
Ethernet	10/100BaseT
I/O	14 PIO, Intel Address/Data-Bus
Supply Voltage	5.0V +/-5%
Active current	300 mA (typical)
Operating temperature (TA)	-25°C to +70°C
Operating system	IPC@CHIP® RTOS
Footprint	DIL32

Features

Six programmable Chip Selects with 256-byte I/O space

Two full-featured asynchronous serial ports allow hardware handshaking with CTS and RTS selectable for each port. Independent baud rate generators.

Flash Memory with min. 10000 cycles

Full integrated 10/100Base-T transceiver (IEEE 802.3 standard)

Traffic LED output

I²C-Bus can be assigned to any PIO pins configured by software

SPI-Bus can be assigned to any PIO pins configured by software

Internal Watchdog

Power detection Input

Two independent DMA channels

Two independent programmable Timer

Five Interrupt inputs, expandable with an external interrupt controller

Pulse Width Demodulator